# Content Editor Reference

# Table of Contents

# 1. Interchange UI −− Content Editor

Interchange is a web presentation framework which has powerful database and content manipulation features. The Interchange Content Editor provides a framework for manipulating pages and components within pages.

## 1.1. Content management concept

Interchange assumes that the *page* is the basic level of content. Each page is typically a file on the filesystem, though it is possible to maintain them in a database instead.

Each page is assumed to be based on a `template`, which is the layout for that page. Each template can contain *slots* for *components*.

Components are interchangeable items that can be inserted in template slots so that pages based on a common template can vary without having to create a separate template.

A *page* may actually be a metacontent template in effect. Examples of this are the flypage for building a product page, and the results page that displays search lists.

## 1.2. A typical page

Here is how the file for a page looks:

```
        [comment]
        ui_name: typical.html
        ui_page_template: leftonly
        [/comment]

        [set page_title] A typical page [/set]
        [set members_only]0[/set]

        [control reset=1]
        [control-set]
                [parameter1]setting1[/parameter1]
                [parameter2]setting2[/parameter2]
        [/control-set]
        [control-set]
                [parameter1]setting1[/parameter1]
                [parameter2]setting2[/parameter2]
        [/control-set]
        [control reset=1]

        @_LEFTONLY_TOP__@

        <!-- BEGIN CONTENT -->

                The content of a typical page.

        <!-- END CONTENT -->

        @_LEFTONLY_BOTTOM__@
```

You can see other examples in the standard Interchange foundation demo.

We will tour the page, adding the PREAMBLE and POSTAMBLE sections.

### Editor information

The comment at the top of the page is only used by the editor to derive things it may need to know. If it were to be removed, the page would display the same; it just may not read and publish the same when edited.

```
[comment]
ui_template_name: leftonly
ui_static: 0
[/comment]
```

In the case above, the template is specified with `ui_template_name` and the page will not be published statically (i.e. `ui_static`: 0).

### PREAMBLE

The PREAMBLE is a section which allows page initialization prior to the header portion of the template, and prior to the setting of the page controls.

```
<!-- BEGIN PREAMBLE -->
        [perl]
                &custom_init();
        [/perl]
<!-- END PREAMBLE -->
```

### Page controls

Page controls are specified in the template definition, and the user can set them from the *page control* menu when editing the page.

```
[set page_title] A typical page [/set]
[set members_only]0[/set]
```

The above sets two page controls, one used in the header to set the HTML title for the page, and the other used to determine if the customer must be logged in (a member) to access the page.

### Component controls

As was said before, a page template can contain multiple components. Each component can have an unlimited number of settings which can be used to specify its behavior. These settings are manipulated in the *component control* sections of the content editor, one for each component slot.

```
[control reset=1]
[control-set]
        [parameter1]setting1[/parameter1]
        [parameter2]setting2[/parameter2]
[/control-set]
[control-set]
        [parameter1]setting1[/parameter1]
        [parameter2]setting2[/parameter2]
[/control-set]
[control reset=1]
```

### Template header section

This is the Variable used to contain the top portion of the template, which may use various *page control* settings and have multiple slots for components.

```
        @_LEFTONLY_TOP__@
```

### CONTENT section

This is the actual content of the page. It can be edited in a TEXTAREA in the content editor.

```
        <!-- BEGIN CONTENT -->

                The content of a typical page.

        <!-- END CONTENT -->
```

### Template footer section

This is the Variable used to contain the bottom portion of the template, which may use various *page control* settings and have multiple slots for components.

```
        @_LEFTONLY_BOTTOM__@
```

### POSTAMBLE

The POSTAMBLE section can contain cleanup code for the page. It will rarely be used.

```
        <!-- BEGIN POSTAMBLE -->
                [perl]
                        &custom_cleanup();
                [/perl]
        <!-- END POSTAMBLE -->
```